



Annotated Sheet Showing
Changes Made

QA Audit

Title	Abbreviation	Chosen
<input type="checkbox"/> Coding Style		<input checked="" type="checkbox"/>
Access Of Static Members Through Objects	AOSMTO	<input checked="" type="checkbox"/>
Assignment To Formal Parameters	ATFP	<input checked="" type="checkbox"/>
Complex Assignment	CA	<input checked="" type="checkbox"/>
Don't Use the Negation Operator Frequently	DUNOF	<input checked="" type="checkbox"/>
Operator '?' May Not Be Used	OMNBU	<input checked="" type="checkbox"/>
Provide Incremental In For-Statement or use w...	PIIFS	<input checked="" type="checkbox"/>
Replacement For Demand Imports	RFDI	<input checked="" type="checkbox"/>
Use Abbreviated Assignment Operator	UAAO	<input checked="" type="checkbox"/>
Use 'this' Explicitly To Access Class Members	UTETACM	<input checked="" type="checkbox"/>
<input type="checkbox"/> Critical Errors		<input checked="" type="checkbox"/>
Avoid Hiding Inherited Attributes	AHIA	<input checked="" type="checkbox"/>
Avoid Hiding Inherited Static Methods	AHISM	<input checked="" type="checkbox"/>
Command Query Separation	CQS	<input checked="" type="checkbox"/>
Hiding Of Names	HON	<input checked="" type="checkbox"/>
Inaccessible Constructor Or Method Matches	ICOMM	<input checked="" type="checkbox"/>
Multiple Visible Declarations With Same Name	MYDWSN	<input checked="" type="checkbox"/>
Overriding a Non-Abstract Method With an Ab...	ONAMWAM	<input checked="" type="checkbox"/>
Overriding a Private Method	OPM	<input checked="" type="checkbox"/>

Severity: High

800
802

Select all Unselect all Set defaults Save set As... Load set...

AOSMTO - Access Of Static Members Through Objects

Static members should be referenced through class names rather than through objects.

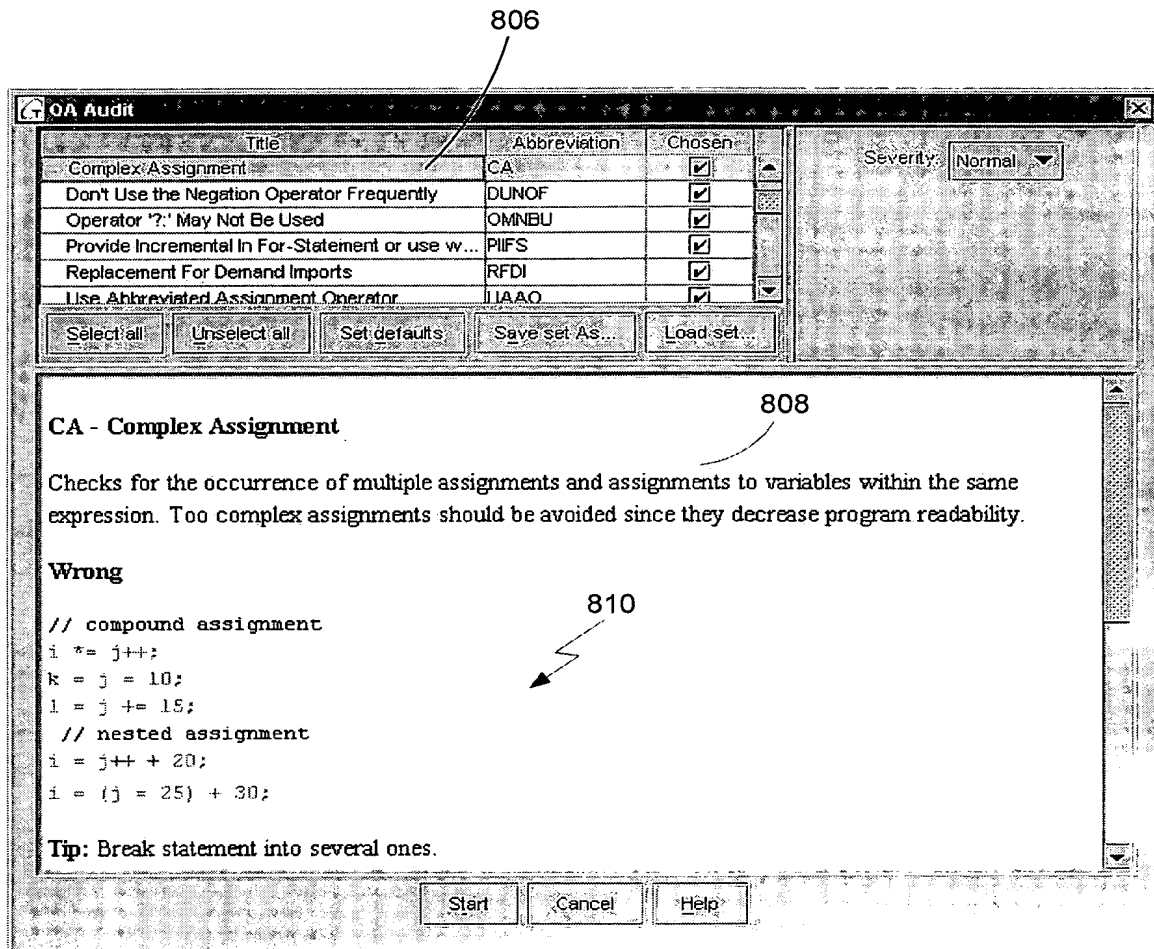
804

Start Cancel Help

Drawing made more clear.
Increased contrast.

FIG. 8A

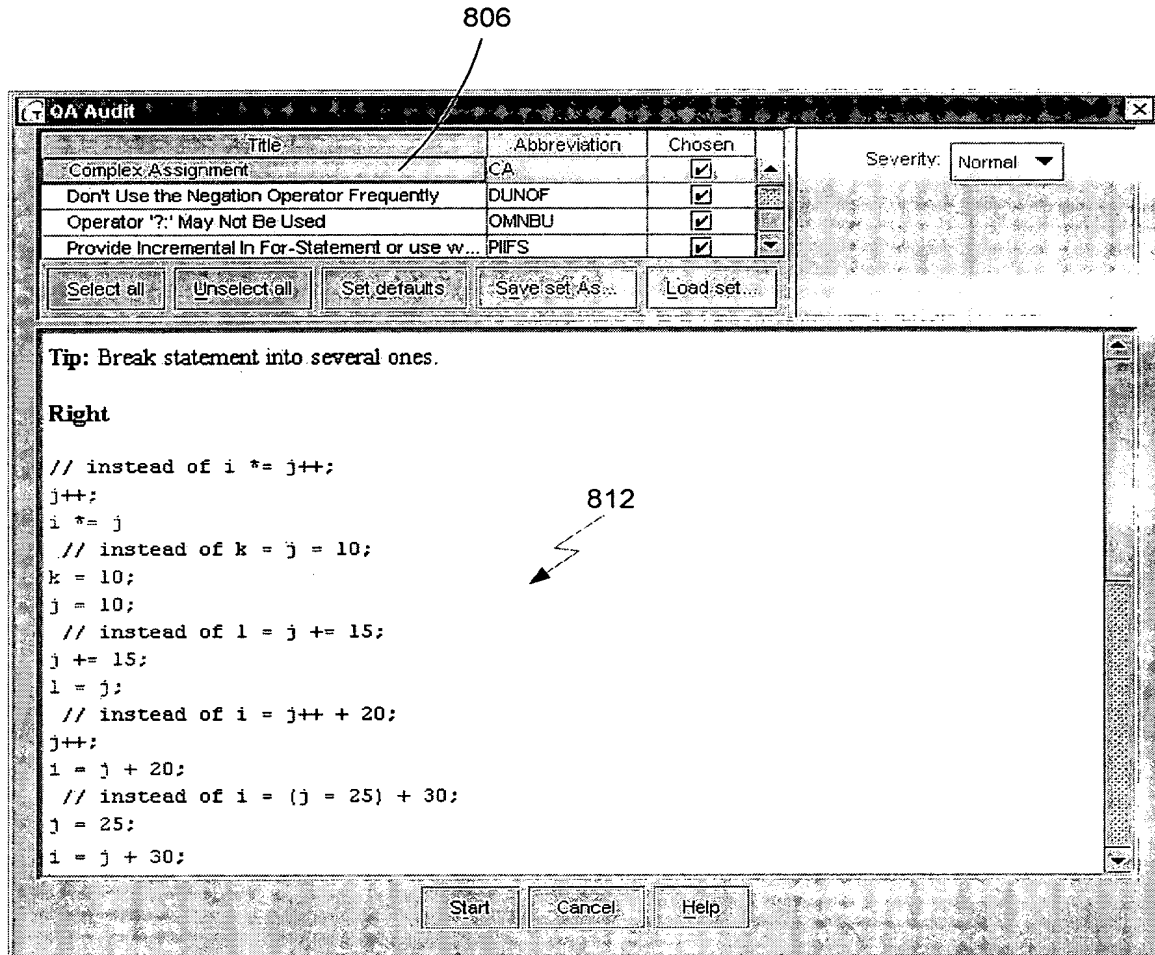
Annotated Sheet Showing
Changes Made



Drawing made more clear.

FIG. 8B

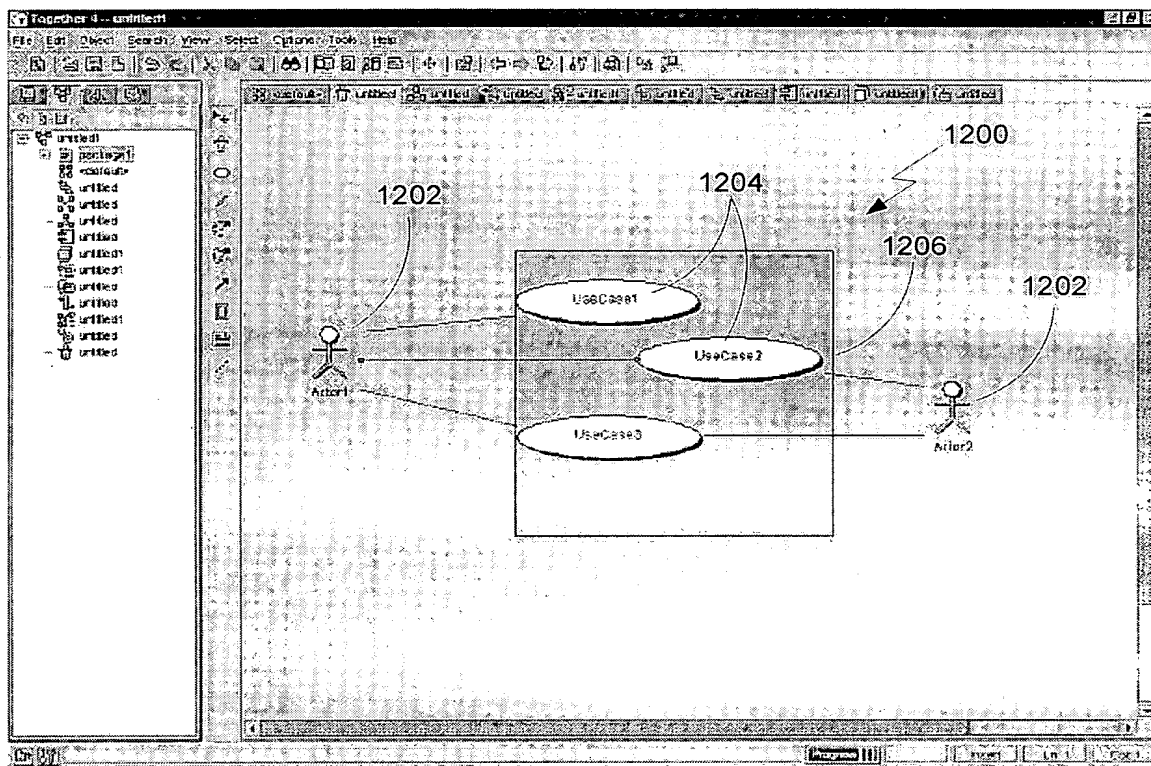
Annotated Sheet Showing
Changes Made



Drawing made more clear.

FIG. 8C

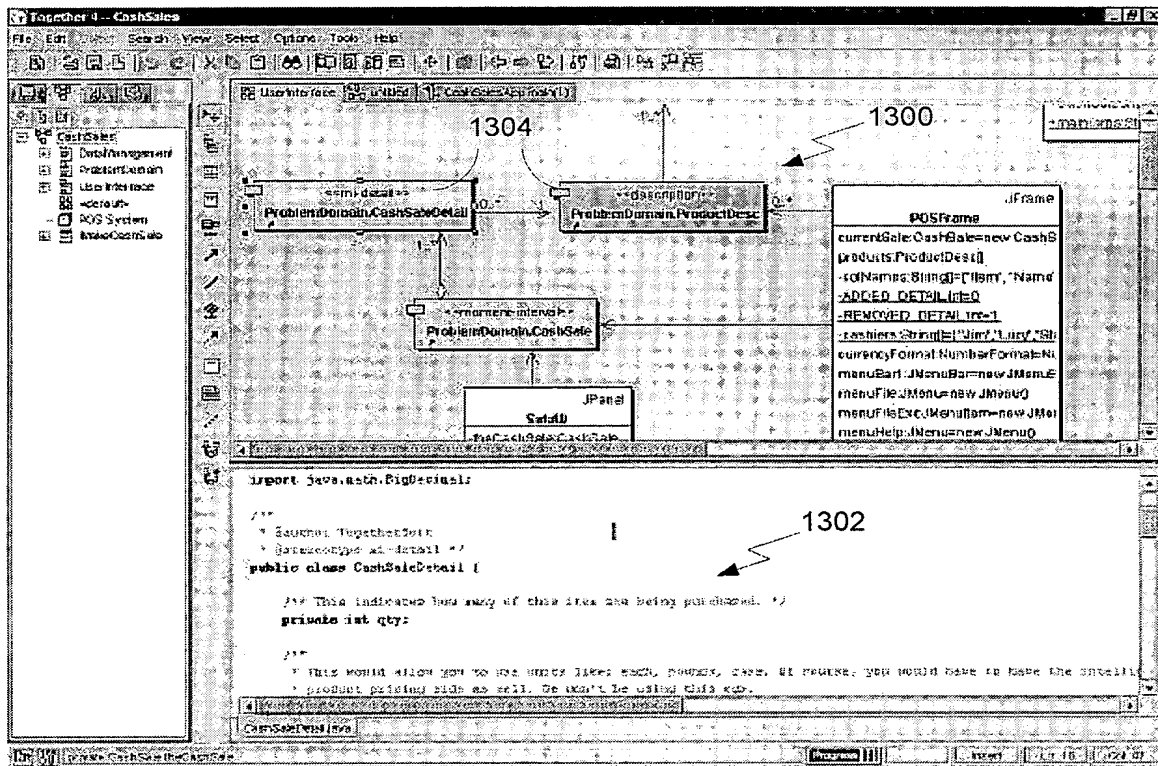
**Annotated Sheet Showing
Changes Made**



Drawing made more clear.

FIG. 12

Annotated Sheet Showing Changes Made



Drawing made more clear.

FIG. 13

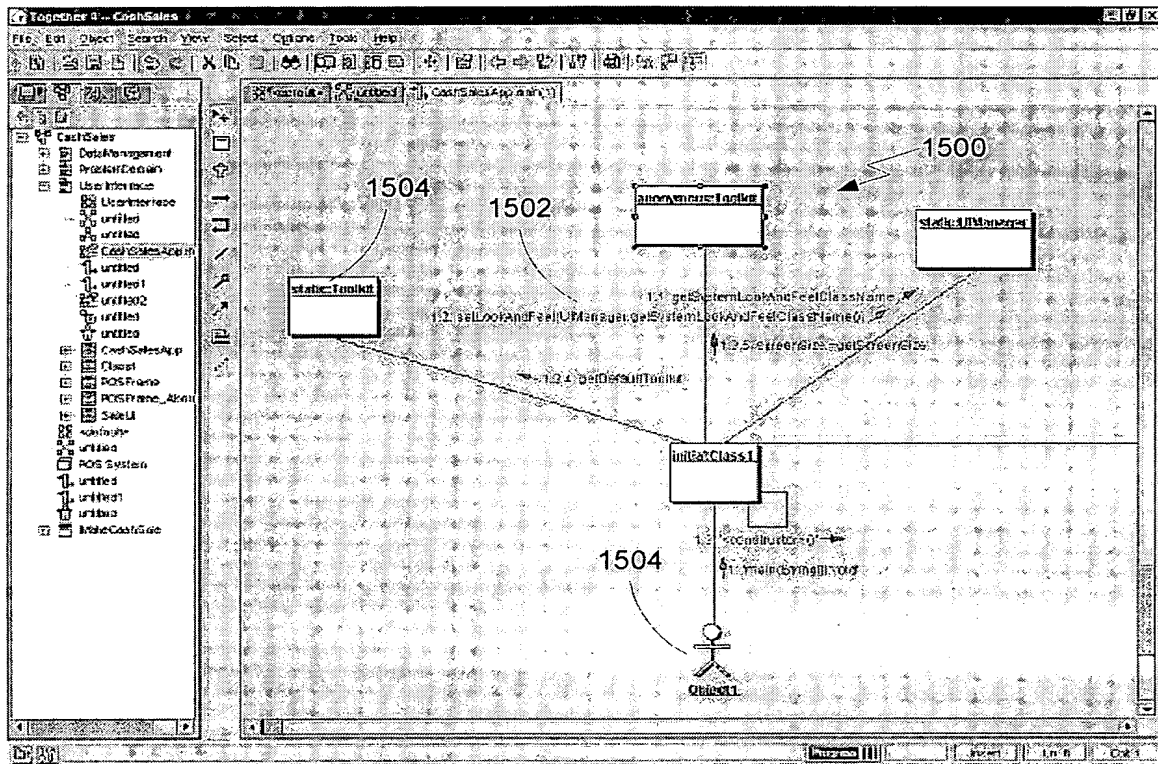
The screenshot displays a UML sequence diagram within a software development environment. The diagram is titled "CashSalesApp.mdi". It features three lifelines: "Client1" (represented by a stick figure), "Initial Class1", and "Store UserManager".

- Client1** initiates the process by calling the `main(String[] args)` method on **Initial Class1**. This call is annotated with a note labeled "1404".
- Initial Class1** then calls the `getSystemLockAndGetLockName` method on **Store UserManager**.
- Subsequently, **Initial Class1** calls the `setLockAndFreeUserManager` method on **Store UserManager**.
- A dashed arrow points from a note labeled "1400" to the **Store UserManager** lifeline.
- At the bottom right, a partial view of another diagram shows a "Name" lifeline with methods like `setName`, `getSystemLockAndGetLockName`, and `setLockAndFreeUserManager`.

The left sidebar of the editor shows a project tree with folders like "CashSales", "Data Manager", and "User Interface", and various files including "CashSalesApp.mdi".

FIG. 14

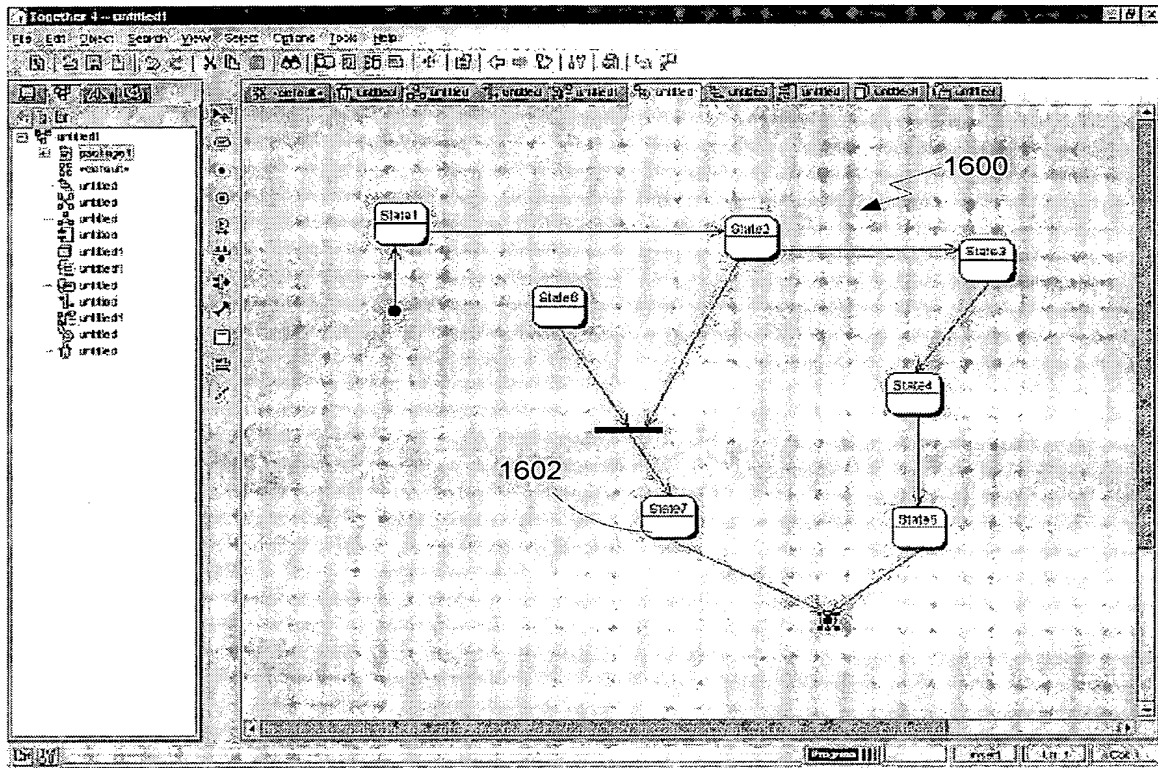
Annotated Sheet Showing
Changes Made



Drawing made more clear.

FIG. 15

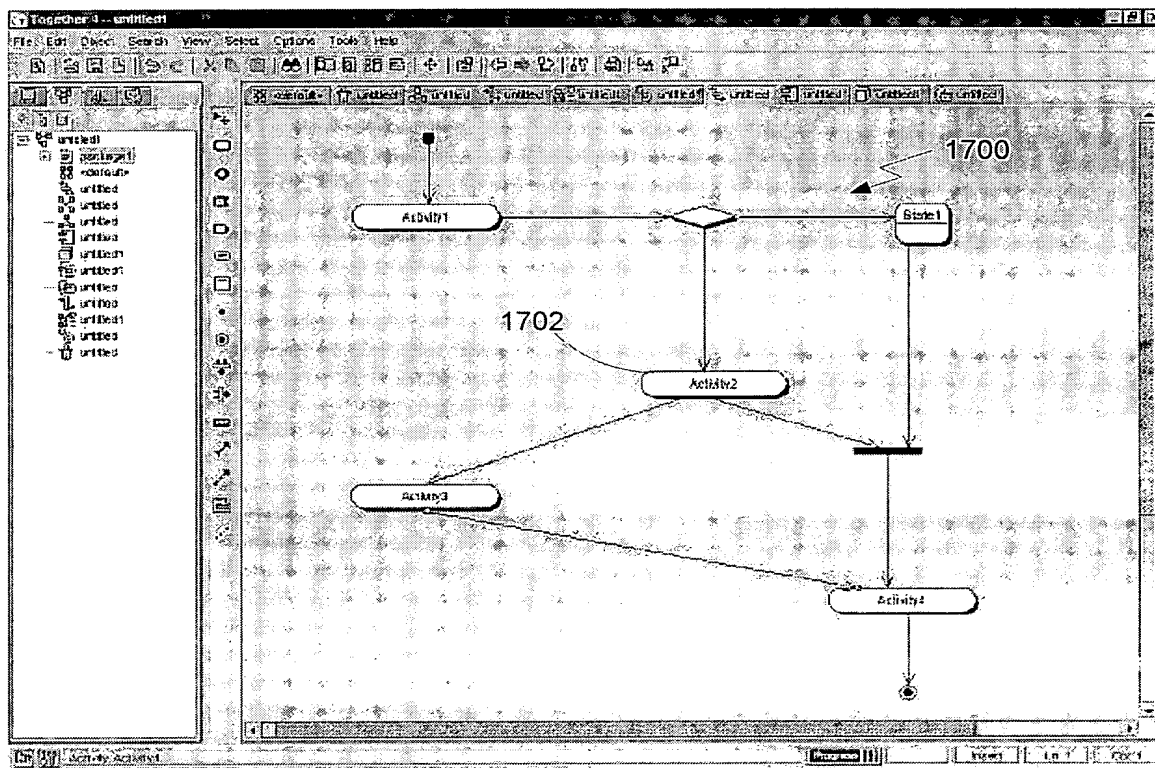
Annotated Sheet Showing
Changes Made



Drawing made more clear.

FIG. 16

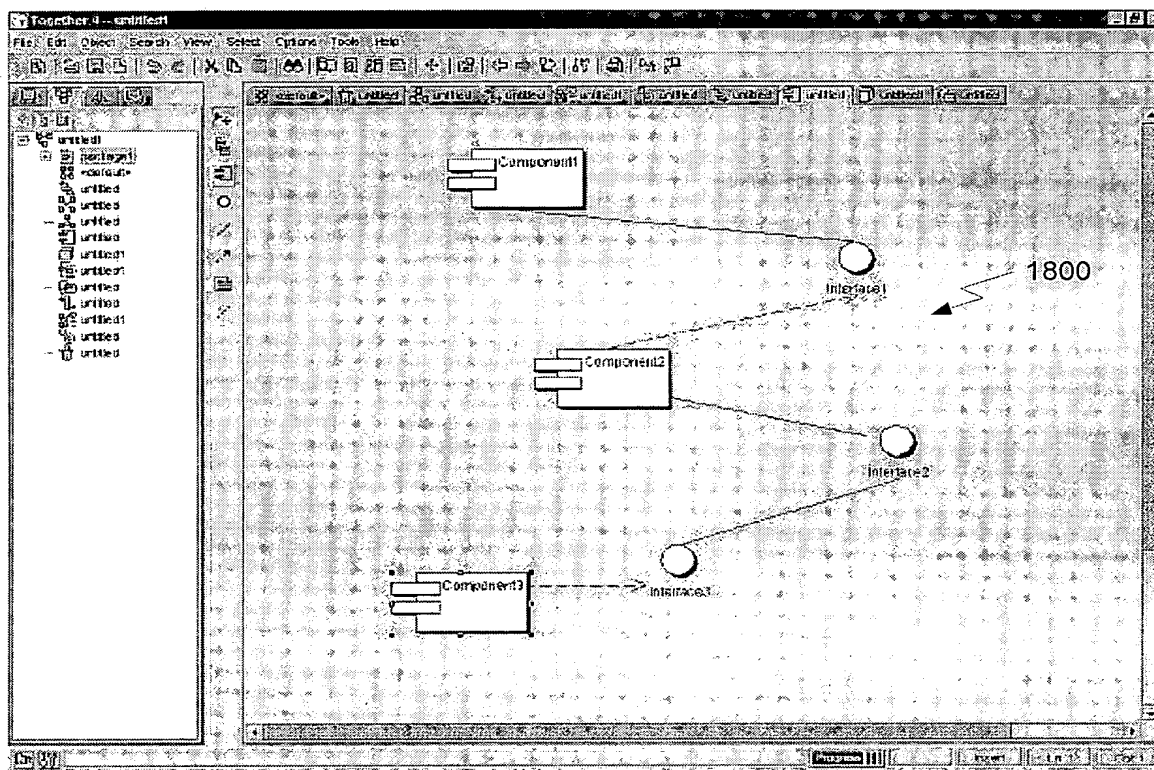
Annotated Sheet Showing
Changes Made



Drawing made more clear.

FIG. 17

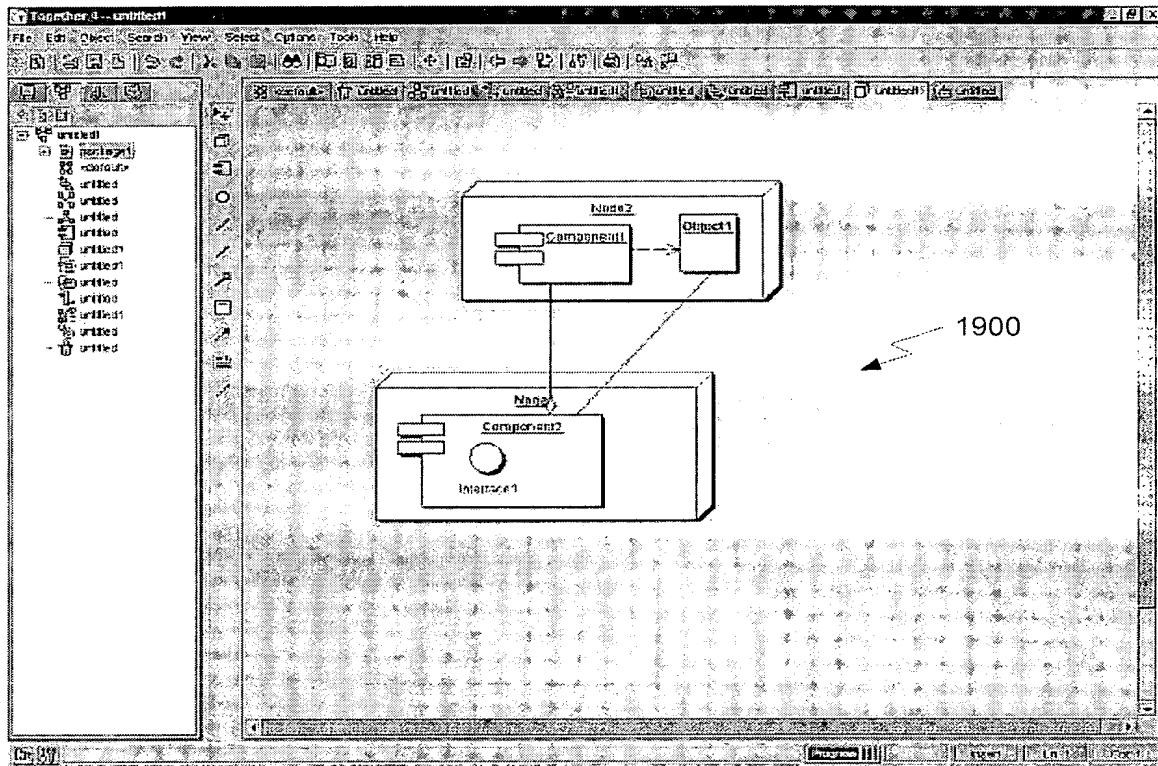
**Annotated Sheet Showing
Changes Made**



Drawing made more clear.

FIG. 18

Annotated Sheet Showing
Changes Made



Drawing made more clear.

FIG. 19